# Weaker Than You Think: A Critical Look at Weakly Supervised Learning

**Dawei Zhu**[1] **Xiaoyu Shen**[2*] **Marius Mosbach**[1] **Andreas Stephan**[3] **Dietrich Klakow**[1]
[1]Saarland University, Saarland Informatics Campus
[2]Amazon Alexa AI
[3]University of Vienna
dzhu@lsv.uni-saarland.de

## Abstract

Weakly supervised learning is a popular approach for training machine learning models in low-resource settings. Instead of requesting high-quality yet costly human annotations, it allows training models with noisy annotations obtained from various weak sources. Recently, many sophisticated approaches have been proposed for robust training under label noise, reporting impressive results. In this paper, we revisit the setup of these approaches and find that *the benefits brought by these approaches are significantly overestimated*. Specifically, we find that the success of existing weakly supervised learning approaches heavily relies on the availability of clean validation samples which, as we show, can be leveraged much more efficiently by simply training on them. After using these clean labels in training, the advantages of using these sophisticated approaches are mostly wiped out. This remains true even when reducing the size of the available clean data to just five samples per class, making these approaches impractical. To understand the true value of weakly supervised learning, we thoroughly analyze diverse NLP datasets and tasks to ascertain when and why weakly supervised approaches work. Based on our findings, we provide recommendations for future research.[1]

## 1 Introduction

Weakly supervised learning (WSL) is one of the most popular approaches for alleviating the annotation bottleneck in machine learning. Instead of collecting expensive clean annotations, it leverages weak labels from various weak labeling sources such as heuristic rules, knowledge bases or lower-quality crowdsourcing (Ratner et al., 2017). These weak labels are inexpensive to obtain, but are often noisy and inherit biases from their sources. Deep learning models trained on such noisy data without regularization can easily overfit to the noisy labels (Zhang et al., 2017; Tänzer et al., 2022). Many advanced WSL techniques have recently been proposed to combat the noise in weak labels, and significant progress has been reported. On certain datasets, they even manage to match the performance of fully-supervised models (Liang et al., 2020; Ren et al., 2020; Yu et al., 2021).

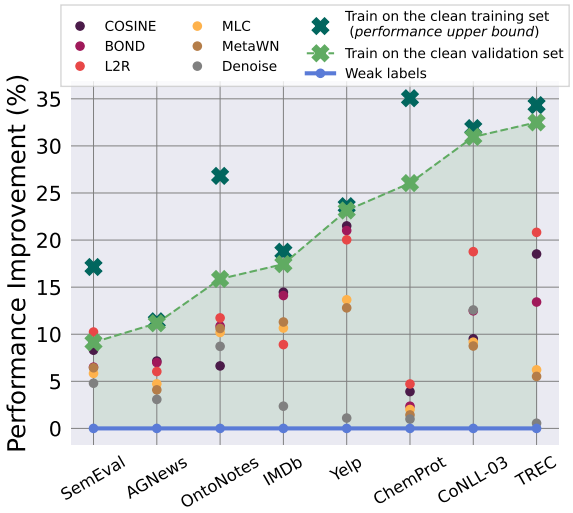



Figure 1: **Performance improvement over weak labels on the test sets.** Each point represents the average performance improvement of one approach over five runs. On various NLP datasets, weakly supervised methods (dots) outperform weak labels (blue line) on the test sets. However, *simply fine-tuning on the available clean validation data (light green crosses) outperforms all sophisticated weakly supervised methods in almost all cases*. See Appendix D.2 for experimental details.

In this paper, we take a close look at the claimed advances of these WSL approaches and find that the *benefits of using them are significantly overestimated*. Although they appear to require only weak labels during training, a substantial number of clean validation samples are used for various purposes such as early-stopping (Liang et al., 2020; Yu et al., 2021) and meta-learning (Ren et al., 2018;

*Work done outside Amazon.
[1]Our code is available at: https://github.com/uds-lsv/critical_wsl



*Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics*
*Volume 1: Long Papers*, pages 14229–14253

Shu et al., 2019; Zheng et al., 2021). We cast doubt on this practice: in real-world applications, these clean validation samples could have instead been used for training. To address our concern, we explore fine-tuning models directly on the validation splits of eight datasets provided by the WRENCH benchmark (Zhang et al., 2021b) and compare it to recent WSL algorithms. The results are shown in Figure 1. Interestingly, although all WSL models generalize better than the weak labels, **simply fine-tuning on the validation splits outperforms all WSL methods in almost all cases**, sometimes even by a large margin. This suggests that existing WSL approaches are not evaluated in a realistic setting and the claimed advances of these approaches may be overoptimistic. In order to determine the true benefits of WSL approaches in a realistic setting, we conduct extensive experiments to investigate the role of clean validation data in WSL. Our findings can be summarized as follows:

- Without access to any clean validation samples, all WSL approaches analyzed in this work *fail to* work, performing similarly to or worse than the weak labels (§4).
- Although increasing the amount of clean validation samples improves WSL performance (§5), these validation samples can be more efficiently leveraged by directly training on them, which can outperform WSL approaches when there are more than 10 samples per class for most datasets (§6).
- Even when enabling WSL models to continue training on clean validation samples, they can barely beat an embarrassingly simple baseline which directly fine-tunes on weak labels followed by fine-tuning on clean samples. This stays true with as few as 5 samples per class (§7).
- The knowledge encoded in pre-trained language models biases them to seek linguistic correlations rather than shallow rules from the weak labels; further fine-tuning the pre-trained language models with contradicting examples helps reduce biases from weak labels (§8).

Altogether, we show that existing WSL approaches significantly overestimate their benefits in a realistic setting. We suggest future work to (1) fully leverage the available clean samples instead of only using them for validation and (2) consider the simple baselines discussed in this work when comparing WSL approaches to better understand WSL's true benefits.

## 2 Related work

**Weak supervision.** Weak supervision is proposed to ease the annotation bottleneck in training machine learning models. It uses weak sources to automatically annotate the data, making it possible to obtain a large amount of annotated data at a low cost. A comprehensive survey is done in Zhang et al. (2022). Ratner et al. (2017) propose to label data programmatically using heuristics such as keywords, regular expressions or knowledge bases. One drawback of weak supervision is that its annotations are noisy, i.e., some annotations are incorrect. Training models on such noisy data may result in poor generalization (Zhang et al., 2017; Tänzer et al., 2022; Zhang et al., 2022). One option to counter the impact of wrongly labeled samples is to re-weight the impact of examples in loss computation (Ren et al., 2018; Shu et al., 2019; Zheng et al., 2021). Another line of research leverages the knowledge encoded in large language models (Ren et al., 2020; Stephan et al., 2022). Methods such as BOND (Liang et al., 2020), ASTRA (Karamanolakis et al., 2021) and COSINE (Yu et al., 2021) apply teacher-student frameworks to train noise-robust models. Zhu et al. (2023) show that teacher-student frameworks may still be fragile in challenging situations and propose incorporating meta-learning techniques in such cases. Multiple benchmarks are available to evaluate weak supervision systems, e.g., WRENCH (Zhang et al., 2021b), Skweak (Lison et al., 2021), and WALNUT (Zheng et al., 2022a). In this paper, we take representative datasets from WRENCH and reevaluate existing WSL approaches in more realistic settings.

**Realistic evaluation.** Certain pitfalls have been identified when evaluating machine learning models developed for low-resource situations. Earlier work in semi-supervised learning (SSL) in computer vision, for example, often trains with a few hundred training examples while retaining thousands of validation samples for model selection (Tarvainen and Valpola, 2017; Miyato et al., 2018). Oliver et al. (2018) criticize this setting and provide specific guidance for realistic SSL evaluation. Recent work in SSL has been adapted to discard the validation set and use a fixed set of hyperpa-

rameters across datasets (Xie et al., 2020; Zhang et al., 2021a; Li et al., 2021). In NLP, it has been shown that certain (prompt-based) few-shot learning approaches are sensitive to prompt selection which requires separate validation samples (Perez et al., 2021). This defeats the purported goal of few-shot learning, which is to achieve high performance even when collecting additional data is prohibitive. Recent few-shot learning algorithms and benchmarks have adapted to a more realistic setting in which fine-grained model selection is either skipped (Gao et al., 2021; Alex et al., 2021; Bragg et al., 2021; Schick and Schütze, 2022; Lu et al., 2022) or the number of validation samples are strictly controlled (Bragg et al., 2021; Zheng et al., 2022b). To our knowledge, no similar work exists exploring the aforementioned problems in the context of weak supervision. This motivates our work.

## 3 Overall setup

**Problem formulation.** Formally, let $\mathcal{X}$ and $\mathcal{Y}$ be the feature and label space, respectively. In standard supervised learning, we have access to a training set $D = \{(x_i, y_i)\}_{i=1}^{N}$ sampled from a clean data distribution $\mathcal{D}_c$ of random variables $(X, Y) \in \mathcal{X} \times \mathcal{Y}$. In weak supervision, we are instead given a weakly labeled dataset $D_w = \{(x_i, \hat{y}_i)\}_{i=1}^{N}$ sampled from a noisy distribution $\mathcal{D}_n$, where $\hat{y}_i$ represents labels obtained from weak labeling sources such as heuristic rules or crowd-sourcing.[2] $\hat{y}_i$ is noisy, i.e., it may be different from the ground-truth label $y_i$. The goal of WSL algorithms is to *obtain a model that generalizes well on* $D_{test} \sim \mathcal{D}_c$ *despite being trained on* $D_w \sim \mathcal{D}_n$. In recent WSL work, a set of clean samples, $D_v \sim \mathcal{D}_c$, is also often included for model selection.[3]

**Datasets.** We experiment with eight datasets covering different NLP tasks in English. Concretely, we include four text classification datasets: (1) AGNews (Zhang et al., 2015), (2) IMDb (Maas et al., 2011), (3) Yelp (Zhang et al., 2015), (4) TREC (Li and Roth, 2002), two relation classification datasets: (5) SemEval (Hendrickx et al., 2010) and (6) ChemProt (Krallinger et al., 2017), and two Named-Entity Recognition (NER) datasets: (7) CoNLL-03 (Tjong Kim Sang and De Meulder, 2003) and (8) OntoNotes (Pradhan et al., 2013). The weak annotations are obtained from the WRENCH (Zhang et al., 2021b) benchmark. Table 1 summarizes the basic statistics of the datasets.

| Dataset | Task | # Class | # Train | # Val | # Test |
|---|---|---|---|---|---|
| AGNews | Topic | 4 | 96K | 12K | 12K |
| IMDb | Sentiment | 2 | 20K | 2.5K | 2.5K |
| Yelp | Sentiment | 2 | 30K | 3.8K | 3.8K |
| TREC | Question | 6 | 4,965 | 500 | 500 |
| SemEval | Relation | 9 | 1,749 | 178 | 600 |
| ChemProt | Relation | 10 | 13K | 1.6K | 1.6K |
| CoNLL-03 | NER | 4 | 14K | 3.2K | 3.4K |
| OntoNotes 5.0 | NER | 18 | 115K | 5K | 23K |

Table 1: **Dataset statistics.** Additional details on datasets are provided in Appendix A.

**WSL baselines.** We analyze popular WSL approaches including: (1) $\mathbf{FT_W}$ represents the standard fine-tuning approach[4] (Howard and Ruder, 2018; Devlin et al., 2019). Ren et al. (2020), Zhang et al. (2021b) and Zheng et al. (2022a) show that a pre-trained language model (PLM) fine-tuned on a weakly labeled dataset often generalizes better than the weak labels synthesized by weak labeling sources. (2) **L2R** (Ren et al., 2018) uses meta-learning to determine the optimal weights for each (noisy) training sample so that the model performs best on the (clean) validation set. Although this method was originally proposed to tackle artificial label noise, we find it performs on par with or better than recent weak supervision algorithms on a range of datasets. (3) **MLC** (Zheng et al., 2021) uses meta-learning as well, but instead of weighting the noisy labels, it uses the meta-model to correct them. The classifier is then trained on the corrected labels. (4) **BOND** (Liang et al., 2020) is a noise-aware self-training framework designed for learning with weak annotations. (5) **COSINE** (Yu et al., 2021) underpins self-training with contrastive regularization to improve noise robustness further and achieves state-of-the-art performance on the WRENCH (Zhang et al., 2021b) benchmark.

To provide a fair comparison, we use RoBERTa-base (Liu et al., 2019) as the common backbone PLM for all WSL approaches (re)implemented in this paper.

[2] Majority voting can be used to resolve conflicting weak labels from different labeling sources.

[3] We refer to model selection as the process of finding the best set of hyperparameters via a validation set, including the optimal early-stopping time. Prior work has shown that early-stopping is crucial for learning with noisy labels (Arpit et al., 2017; Yu et al., 2021; Zhu et al., 2022; Tänzer et al., 2022).

[4] We use the subscript "W" to emphasize that this fine-tuning is done on the weakly annotated data and to distinguish it from the fine-tuning experiments in Section 6 which are done on clean data.

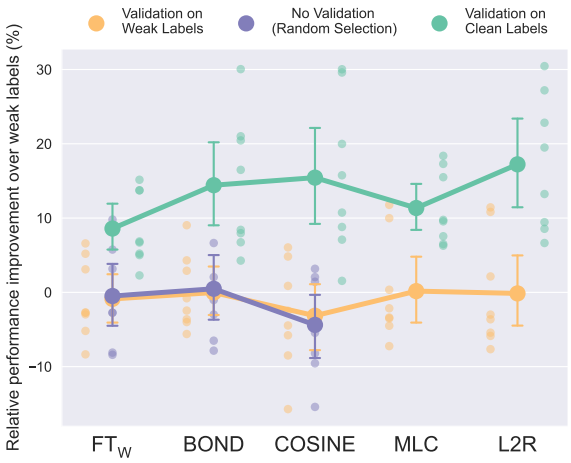



Figure 2: **Relative performance gain over weak labels when varying validation conditions.** The dots show the average performance gain across 5 runs for each of the 8 datasets. The curves show the average gain across datasets. WSL baselines achieve noticeable performance gains only if a clean validation set is used. Performing model selection on a weakly labeled validation set does not help generalization. Note that L2R and MLC are not applicable without validation data.

## 4 Is clean data necessary for WSL?

Recent best-performing WSL approaches rely on a clean validation set for model selection. Figure 1 reveals that they fail to outperform a simple model that is directly fine-tuned on the validation set. Therefore, a natural question to ask is: "*Will WSL still work without accessing the clean validation set?*". If the answer is yes, then we can truly reduce the burden of data annotation and the benefits of these WSL approaches would be undisputed. This section aims to answer this question.

**Setup.** We compare three different validation choices for model selection using either (1) a clean validation set from $D_v$ as in prior work, (2) weak labels from $\tilde{D}_v$ obtained by annotating the validation set via weak labeling sources (the same procedure used to construct training annotations), or (3) no validation data at all. In the last setting, we randomly select 5 sets of hyperparameters from our search space (see Appendix C). We run the WSL approaches introduced in Section 3 on all eight datasets with different validation choices and measure their test performance. Each experiment is repeated 5 times with different seeds.

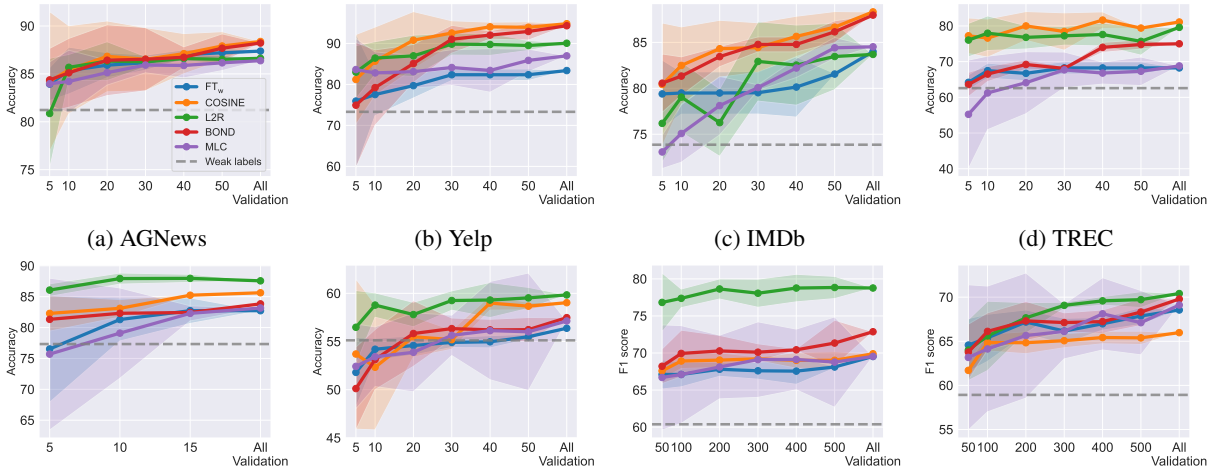
While one may expect a certain drop in performance when switching from $D_v$ to $\tilde{D}_v$, the absolute performance of a model does not determine the usefulness of a WSL method. We are more interested in whether a trained model generalizes better than the weak labels.[5] In realistic applications, it is only worth deploying trained models if they demonstrate clear advantages over the weak labels. Therefore, we report the relative performance gain of WSL approaches over the weak labels. Formally, let $P_{WL}, P_\alpha$ denote the performance (accuracy, F1-score, etc.) achieved by the weak labels and a certain WSL method $\alpha$, respectively. The the relative performance gain is defined as $G_\alpha = (P_\alpha - P_{WL})/P_{WL}$. We consider a WSL approach to be *effective* and practically useful only if $G_\alpha > 0$.

**Results.** Figure 2 shows the relative performance gain for all considered WSL approaches. When model selection is performed on a clean validation set (green curve), all weak supervision baselines generalize better than the weak labels. Sophisticated methods like COSINE and L2R push the performance even further. This observation is consistent with previous findings (Zhang et al., 2021b; Zheng et al., 2022a). However, when using a weakly labeled validation set (yellow curve), all WSL baselines become *ineffective* and barely outperform the weak labels. More interestingly, models selected through the weakly labeled validation sets do not outperform models configured with random hyperparameters (purple curve). These results demonstrate that model selection on clean validation samples plays a vital role in the effectiveness of WSL methods. **Without clean validation samples, existing WSL approaches do not work.**

## 5 How much clean data does WSL need?

Now that we know clean samples are necessary for WSL approaches to work, a follow-up question would be: "How many clean samples do we need?" Intuitively, we expect an improvement in performance as we increase the amount of clean data, but it is unclear how quickly this improvement starts to level off, i.e., we may find that a few dozen clean samples are enough for WSL approaches to perform model selection. The following section seeks to answer this question.

[5] Weak labeling sources are typically applied to the training data to synthesize a weakly annotated training set. However, it is also possible to synthesize the weak labels for the test set following the same procedure and measure their performance. In other words, weak labeling sources can be regarded as the most basic classification model, and the synthesized weak labels are its predictions.




(a) AGNews (b) Yelp (c) IMDb (d) TREC

(e) SemEval (f) ChemProt (g) CoNLL-03 (h) OntoNotes 5.0

Figure 3: **The impact of the number of clean validation samples on performance.** We plot average performance and standard deviation over 5 runs varying the size of the clean validation data. Whenever a small proportion of validation data is provided, most WSL techniques generalize better than the weak label baseline (grey dashed line). Performance improves with additional validation samples, but this tendency usually levels out with a moderate number of validation samples.

**Setup.** We apply individual WSL approaches and vary the size of clean data sub-sampled from the original validation split. For text and relation classification tasks, we draw an increasing number of clean samples $N \in \{5, 10, 15, 20, 30, 40, 50\}$ per class when applicable.[6] In the case of NER, as a sentence may contain multiple labels from different classes, selecting exactly $N$ samples per class at random is impractical. Hence, for NER we sample $N \in \{50, 100, 200, 300, 400, 500\}$ sentences for validation. For each $N$, we run the same experiment 5 times. Note that the clean data is *used solely for model selection* in this set of experiments.

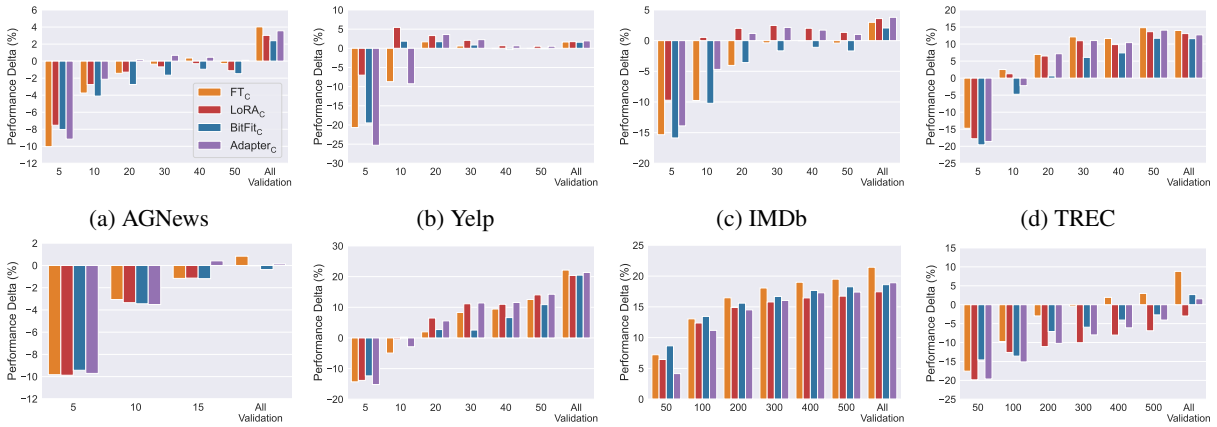
**Results.** As shown in Figure 3, in most cases, a handful of validation samples already make WSL work better than the weak labels. We observe an increasing trend in performance with more validation samples, but typically this trend weakens with a moderate size of samples (~30 samples per class or ~200 sentences) and adding more samples provides little benefit. There are a few exceptions. For example, on IMDb all methods except L2R consistently perform better with more validation data. On CoNLL-03, on the other hand, most methods seem to be less sensitive to the number of samples. Overall, the results suggest that **a small amount of clean validation samples may be sufficient for current WSL methods to achieve good performance.** Using thousands of validation samples, like in the established benchmarks (Zhang et al., 2021b; Zheng et al., 2022a), is neither realistic nor necessary.

[6]The validation set of SemEval is too small to support $N > 20$. Also, if a dataset is unbalanced, we randomly select $N \times C$ samples, where $C$ denotes the number of classes. This is a realistic sampling procedure when performing data annotation.

## 6 Is WSL useful with less clean data?

The previous sections have shown that current WSL approaches (1) do not improve over direct fine-tuning on the existing validation splits (Figure 1) and (2) require only a small amount of validation samples to be effective (Figure 3). This section investigates whether the conclusion from Figure 1 would change with less clean data, i.e., can WSL approaches outperform direct fine-tuning when less clean data is available?

**Setup.** We follow the same procedure as in Section 5 to subsample the *cleanly annotated* validation sets and fine-tune models directly on the sampled data. In addition to the standard fine-tuning approach (Devlin et al., 2019), we also experiment with three parameter-efficient fine-tuning (PEFT) approaches as – in the few-shot setting – they have been shown to achieve comparable or even better performance than fine-tuning all parameters (Peters et al., 2019; Logan IV et al., 2022; Liu et al., 2022). In particular, we include adapters (Houlsby et al., 2019), LoRA (Hu et al., 2022), and BitFit (Zaken et al., 2022).




(a) AGNews (b) Yelp (c) IMDb (d) TREC

(e) SemEval (f) ChemProt (g) CoNLL-03 (h) OntoNotes 5.0

Figure 4: **Using clean data for validation vs. training.** We show the average performance (Acc. and F1-score in %) difference between (parameter-efficient) fine-tuning approaches and COSINE when varying amounts of clean samples. COSINE uses the clean samples for validation, whereas fine-tuning approaches directly train on them (indicated in the legend with the subscript 'C'). For most sequence classification tasks, fine-tuning approaches work better once 10 clean samples are available for training. For NER, several hundreds of clean sentences may be required to attain better results via fine-tuning. Refer to Appendix D for a comparison with other WSL approaches.

We use one fixed set of hyperparameter configurations and train models for 6000 steps on each dataset.[7] We report performance at the last step and compare it with WSL approaches which use the same amount of clean data for validation.

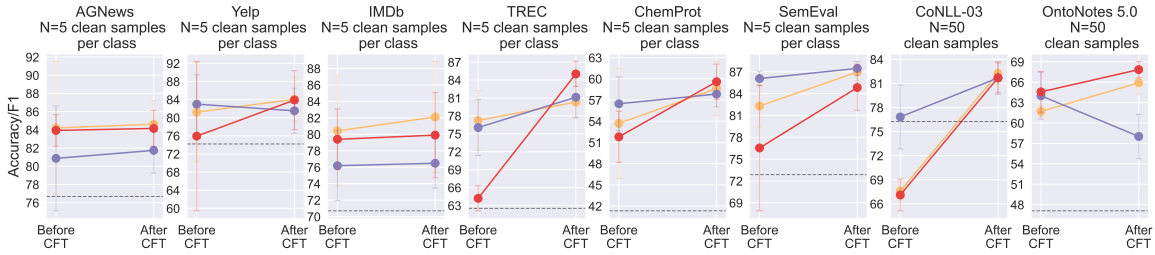
**Results.** Figure 4 shows the performance difference between the fine-tuning baselines and COSINE, one of the best-performing WSL approaches, when varying the number of clean samples. It can be seen that in extremely low-resource cases (less than 5 clean samples per class), COSINE outperforms fine-tuning. However, fine-tuning approaches quickly take over when more clean samples are available. LoRA performs better than COSINE on three out of four text classification tasks with just 10 samples per class. AGNews is the only exception, where COSINE outperforms LoRA by about 1% when 20 samples per class are available, but adapters outperform COSINE in this case. Relation extraction has the same trend where 10–20 samples per class are often enough for fine-tuning approaches to catch up. For NER tasks, all fine-tuning approaches outperform COSINE with as few as 50 sentences on CoNLL-03. OntoNotes seems to be more challenging for fine-tuning and 400 sentences are required to overtake COSINE. Still, 400 sentences only account for 0.3% of the weakly labeled samples used for training COSINE. This indicates that models can benefit much more from training on a small set of clean data rather than on vast amounts of weakly labeled data. Note that the fine-tuning approaches we experiment with work out-of-the-box across NLP tasks. If one specific task is targeted, few-shot learning methods with manually designed prompts might perform even better.[8] Hence, the performance shown here should be understood as a lower bound of what one can achieve by fine-tuning. Nevertheless, we can see that even considering the lower bound of fine-tuning-based methods, **the advantage of using WSL approaches vanishes when we have as few as 10 clean samples per class**. For many real-world applications, this annotation workload may be acceptable, limiting the applicability of WSL approaches.

## 7 Can WSL benefit from fine-tuning?

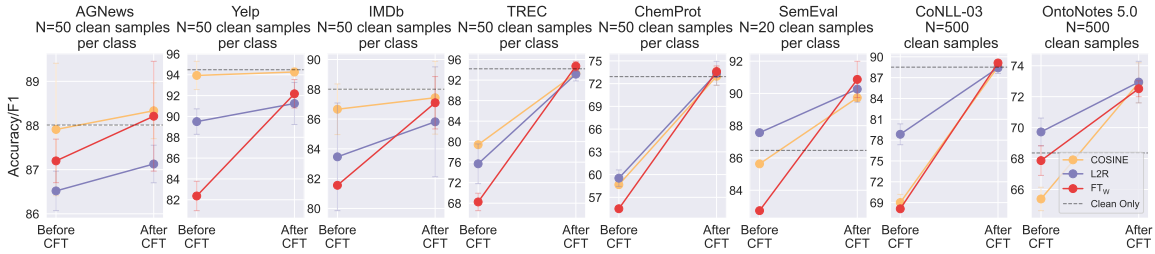
The WSL approaches have only used clean samples for validation so far, which is shown to be inefficient compared to training directly on them.

[7]The hyperparameters are randomly picked from the ranges mentioned in the original papers of corresponding methods and fixed across all experiments. *We did not cherry-pick them based on the test performances*. In most cases the training loss converges within 300 steps. We intentionally extend training to show that we do not rely on extra data for early-stopping. We find that overfitting to the clean data does not hurt generalization. A similar observation is made in Mosbach et al. (2021). Detailed configurations are presented in Appendix D.

[8]For example, Zhao et al. (2021) achieve an accuracy of 85.9% on AGNews using just 4 labeled samples in total. For comparison, COSINE needs 20 labeled samples for validation to reach 84.21%.




(a) $N = 5$ clean samples per class for classification tasks. $N = 50$ clean samples for NER tasks.




(b) $N = 50$ clean samples per class for classification tasks except for SemEval due to its limited validation size. $N = 500$ clean samples for NER tasks

Figure 5: **Performance before and after continuous fine-tuning (CFT) on the clean data**. The average performance and standard deviation over 5 runs are reported. Though CFT improves the performance of WSL approaches in general, the simplest baseline $\text{FT}_\text{W}$ gains the most from it. After applying CFT, $\text{FT}_\text{W}$ performs on par with or better than more sophisticated WSL approaches, suggesting these sophisticated approaches might have overestimated their actual value. Further plots are included in Appendix F.

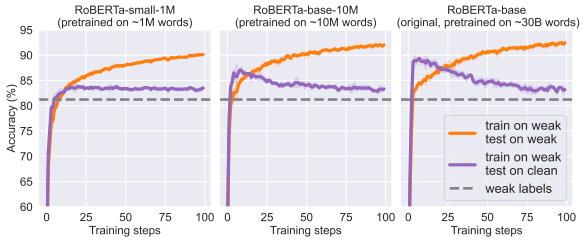
We question whether enabling WSL methods to further fine-tune on these clean samples would improve their performance. In this section, we study a straightforward training approach that makes use of both clean and weak labels.[9]

**Setup.** Given both the weakly labeled training data and a small amount of clean data, we consider a simple two-phase training baseline. In the first phase, we apply WSL approaches on the weakly labeled training set, using the clean data for validation. In the second phase, we take the model trained on the weakly labeled data as a starting point and continue to train it on the clean data. We call this approach continuous fine-tuning (**CFT**). In our experiment, we apply CFT to the two best-performing WSL approaches, COSINE and L2R, along with the most basic WSL baseline, $\text{FT}_\text{W}$. We sample clean data in the same way as in Section 5. The training steps of the second phase are fixed at 6000. Each experiment is repeated 5 times with different seeds.

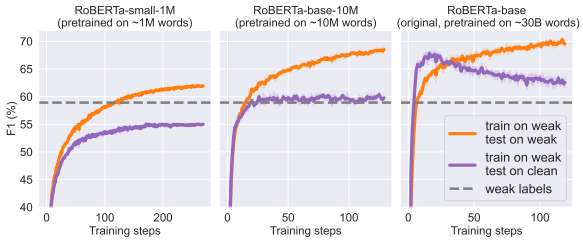
**Results.** Figure 5 shows the model performance before and after applying CFT. It can be seen that CFT does indeed benefit WSL approaches in most cases even with very little clean data (Figure 5a). For L2R, however, the improvement is less obvious, and there is even a decrease on Yelp and OntoNotes. This could be because L2R uses the validation loss to reweight training samples, meaning that the value of the validation samples beyond that may only be minimal. When more clean samples are provided, CFT exhibits a greater performance gain (Figure 5b). It is also noticeable that CFT reduces the performance gap among all three WSL methods substantially. Even the simplest approach, $\text{FT}_\text{W}$, is comparable to or beats L2R and COSINE in all tasks after applying CFT. Considering that COSINE and L2R consume far more computing resources, our findings suggest that **the net benefit of using sophisticated WSL approaches may be significantly overestimated and impractical for real-world use cases.**

Finally, we find the advantage of performing WSL diminishes with the increase of clean samples even after considering the boost from CFT. When 50 clean samples per class (500 sentences for NER) are available, applying WSL+CFT only results in a performance boost of less than 1% on 6 out of 8 datasets, compared with the baseline which only fine-tunes on clean samples. Note that weak la-

[9] In Appendix E we also explored other baselines that combine clean and weak data, but they perform considerably worse than the approach we consider in this section.

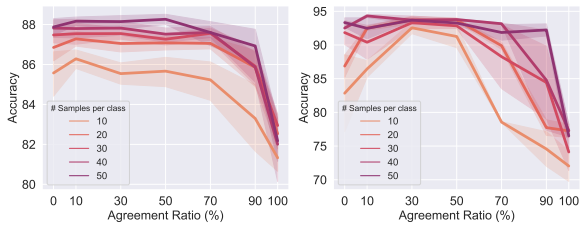



(a) AGNews

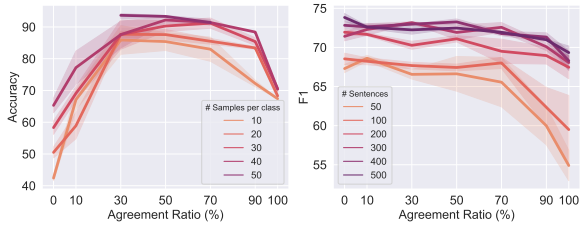



(b) OntoNotes 5.0

Figure 6: **Performance curves of different PLMs during training**. PLMs are trained on weak labels and evaluated on both clean and weakly labeled test sets. Pre-training on larger corpora improves performance on the clean distribution. Further plots are in Appendix G.




(a) AGNews (b) Yelp




(c) TREC* (d) OntoNotes 5.0

Figure 7: **Model performance varying the number of clean samples $N$ and agreement ratio $\alpha$.** Large $\alpha$ generally causes a substantial drop in performance. *: Certain combinations of $\alpha$ and $N$ are not feasible because the validation set lacks samples with clean and weak labels that coincide or differ. Further plots are in Appendix G.

bels are no free lunch. Managing weak annotation resources necessitates experts who not only have linguistic expertise for annotation but also the ability to transform that knowledge into programs to automate annotations. This additional requirement naturally reduces the pool of eligible candidates and raises the cost. In this situation, annotating a certain amount of clean samples may be significantly faster and cheaper. Thus, we believe WSL has a long way to go before being truly helpful in realistic low-resource scenarios.

## 8 What makes $\text{FT}_\text{W}$+CFT effective?

As seen in the previous section, combining $\text{FT}_\text{W}$ with CFT yields a strong baseline that more sophisticated WSL approaches can hardly surpass. This section examines factors that contribute to the effectiveness of this method. Specifically, we aim to answer two questions: (1) "How does $\text{FT}_\text{W}$ resist biases despite being trained only on weak labels?" and (2) "How does CFT further reduce bias introduced by weak labels?".

**Setup.** To answer question (1), we modify the backbone PLM to see if its encoded knowledge plays an important role. We explore two additional PLMs that are pre-trained on less data: RoBERTa-small-1M and RoBERTa-base-10M, which are pre-trained on 1M and 10M words, respectively.[10] We report model performance on both clean labels and weak labels to see which labels the model tends to fit. To answer question (2), we vary the agreement ratio in the clean samples to see how these clean labels help combat biases from weak labels. The agreement ratio is defined as the percentage of samples whose clean labels match the corresponding weak labels. Intuitively, if the clean label for a specific training example matches its weak label, then this example may not contribute additional information to help combat bias. A higher agreement ratio should therefore indicate fewer informative samples.

**Results.** Figure 6 shows the performances for different PLMs. Pre-training on more data clearly helps to overcome biases from weak labels. When the pre-training corpus is small, the model tends to fit the noisy weak labels more quickly than the clean labels and struggles to outperform weak labels throughout the entire training process. With a large pre-training corpus, however, the model can make better predictions on clean labels than

[10]The original RoBERTa-base model is pre-trained on 100B words. The two less pre-trained models are obtained from (Warstadt et al., 2020). RoBERTa-base-10M retains the same architecture as RoBERTa-base, while RoBERTa-small-1M contains fewer parameters.

weak labels in the early stages of training, even when it is only trained on weak labels. If we apply proper early-stopping before the model is eventually dragged toward weak labels, we can attain a model that generalizes significantly better than the weak labels. This indicates that *pre-training provides the model with an inductive bias to seek more general linguistic correlations instead of superficial correlations from the weak labels*, which aligns with previous findings in Warstadt et al. (2020). This turns out to be the key to why simple $\text{FT}_\text{W}$ works here. Figure 7 shows how the agreement ratio $\alpha$ in clean samples affects the performance. Performance declines substantially for $\alpha > 70\%$, showing that it is necessary to have contradictory samples in order to reap the full advantage of CFT. This is reasonable, given that having examples with clean labels that coincide with their weak labels may reinforce the unintended bias learned from the weakly labeled training set. The optimal agreement ratio lies around 50%. However, having $\alpha = 0$ also yields decent performance for most datasets except TREC, suggesting contradictory samples play a more important role here and at least a minimum set of contradictory samples are required for CFT to be beneficial.

## 9 Conclusions and recommendations

Our extensive experiments provide strong evidence that recent WSL approaches heavily overestimate their performance and practicality. We demonstrated that they hinge on clean samples for model selection to reach the claimed performance, yet models that are simply trained on these clean samples are already better. When both clean and weak labels are available, a simple baseline ($\text{FT}_\text{W}$+CFT) performs on par with or better than more sophisticated methods while requiring much less computation and effort for model selection.

Inspired by prior work (Oliver et al., 2018; Perez et al., 2021), our recommendations for future WSL approaches are the following:

- Report the model selection criteria for proposed methods and, especially, how much they rely on the presence of clean data.
- Report how many cleanly annotated samples are required for a few-shot learning approach to reach the performance of a proposed WSL approach. If thousands of weakly annotated samples are comparable to a handful of clean samples – as we have seen in Section 6 – then WSL may not be the best choice for the given low-resource setting.
- If a proposed WSL method requires extra clean data, such as for validation, then the simple $\text{FT}_\text{W}$+CFT baseline should be included in evaluation to claim the real benefits gained by applying the method.

We hope our findings and recommendations will spur more robust future work in WSL such that new methods are truly beneficial in realistic low-resource scenarios.

## Limitations

We facilitate fair comparisons and realistic evaluations of recent WSL approaches. However, our study is not exhaustive and has the following limitations.

First, it may be possible to perform model selection by utilizing prior knowledge about the dataset. For example, if the noise ratio (the proportion of incorrect labels in the training set) is known in advance, it can be used to determine (a subset of) hyperparameters (Han et al., 2018; Li et al., 2020). In this case, certain WSL approaches may still work without access to extra clean data.

Second, in this paper we concentrate on tasks in English where strong PLMs are available. As we have shown in Section 6, training them on a small amount of data is sufficient for generalization. For low-resource languages where no PLMs are available, training may not be that effective, and WSL methods may achieve higher performance.

Third, we experiment with datasets from the established WRENCH benchmark, where the weak labels are frequently assigned by simple rules like as regular expressions (see Appendix B for examples). However, in a broader context, weak supervision can have different forms. For example, Smith et al. (2022) generates weak labels through large language models. Zhou et al. (2022) use hyper-link information as weak labels for passage retrieval. We have not extended our research to more diverse types of weak labels.

Despite the above limitations, however, we identify the pitfalls in the existing evaluation of current WSL methods and demonstrate simple yet strong baselines through comprehensive experiments on a wide range of tasks.

## Acknowledgments

We thank Vagrant Gautam for thoughtful suggestions and insightful discussions. We would also like to thank our anonymous reviewers for their constructive feedback.

This work has been partially funded by the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) – Project-ID 232722074 – SFB 1102 and the EU Horizon 2020 projects ROXANNE under grant number 833635.

## References

Neel Alex, Eli Lifland, Lewis Tunstall, Abhishek Thakur, Pegah Maham, C. Jess Riedel, Emmie Hine, Carolyn Ashurst, Paul Sedille, Alexis Carlier, Michael Noetel, and Andreas Stuhlmüller. 2021. RAFT: A real-world few-shot text classification benchmark. In *Proceedings of the Neural Information Processing Systems Track on Datasets and Benchmarks 1, NeurIPS Datasets and Benchmarks 2021, December 2021, virtual*.

Devansh Arpit, Stanislaw Jastrzebski, Nicolas Ballas, David Krueger, Emmanuel Bengio, Maxinder S. Kanwal, Tegan Maharaj, Asja Fischer, Aaron C. Courville, Yoshua Bengio, and Simon Lacoste-Julien. 2017. A closer look at memorization in deep networks. In *Proceedings of the 34th International Conference on Machine Learning, ICML 2017, Sydney, NSW, Australia, 6-11 August 2017*, volume 70 of *Proceedings of Machine Learning Research*, pages 233–242. PMLR.

Jonathan Bragg, Arman Cohan, Kyle Lo, and Iz Beltagy. 2021. FLEX: unifying evaluation for few-shot NLP. In *Advances in Neural Information Processing Systems 34: Annual Conference on Neural Information Processing Systems 2021, NeurIPS 2021, December 6-14, 2021, virtual*, pages 15787–15800.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2019, Minneapolis, MN, USA, June 2-7, 2019, Volume 1 (Long and Short Papers)*, pages 4171–4186. Association for Computational Linguistics.

Tianyu Gao, Adam Fisch, and Danqi Chen. 2021. Making pre-trained language models better few-shot learners. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing, ACL/IJCNLP 2021, (Volume 1: Long Papers), Virtual Event, August 1-6, 2021*, pages 3816–3830. Association for Computational Linguistics.

Bo Han, Quanming Yao, Xingrui Yu, Gang Niu, Miao Xu, Weihua Hu, Ivor W. Tsang, and Masashi Sugiyama. 2018. Co-teaching: Robust training of deep neural networks with extremely noisy labels. In *Advances in Neural Information Processing Systems 31: Annual Conference on Neural Information Processing Systems 2018, NeurIPS 2018, December 3-8, 2018, Montréal, Canada*, pages 8536–8546.

Iris Hendrickx, Su Nam Kim, Zornitsa Kozareva, Preslav Nakov, Diarmuid Ó Séaghdha, Sebastian Padó, Marco Pennacchiotti, Lorenza Romano, and Stan Szpakowicz. 2010. SemEval-2010 task 8: Multi-way classification of semantic relations between pairs of nominals. In *Proceedings of the 5th International Workshop on Semantic Evaluation*, pages 33–38, Uppsala, Sweden. Association for Computational Linguistics.

Neil Houlsby, Andrei Giurgiu, Stanislaw Jastrzebski, Bruna Morrone, Quentin de Laroussilhe, Andrea Gesmundo, Mona Attariyan, and Sylvain Gelly. 2019. Parameter-efficient transfer learning for NLP. In *Proceedings of the 36th International Conference on Machine Learning, ICML 2019, 9-15 June 2019, Long Beach, California, USA*, volume 97 of *Proceedings of Machine Learning Research*, pages 2790–2799. PMLR.

Jeremy Howard and Sebastian Ruder. 2018. Universal language model fine-tuning for text classification. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics, ACL 2018, Melbourne, Australia, July 15-20, 2018, Volume 1: Long Papers*, pages 328–339. Association for Computational Linguistics.

Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. 2022. LoRA: Low-rank adaptation of large language models. In *The Tenth International Conference on Learning Representations, ICLR 2022, Virtual Event, April 25-29, 2022*. OpenReview.net.

Giannis Karamanolakis, Subhabrata Mukherjee, Guoqing Zheng, and Ahmed Hassan Awadallah. 2021. Self-training with weak supervision. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2021, Online, June 6-11, 2021*, pages 845–863. Association for Computational Linguistics.

Martin Krallinger, Obdulia Rabal, Saber A Akhondi, Martın Pérez Pérez, Jesús Santamaría, Gael Pérez Rodríguez, Georgios Tsatsaronis, Ander Intxaurrondo, José Antonio López, Umesh Nandal, et al. 2017. Overview of the BioCreative VI chemical-protein interaction track. In *Proceedings of the sixth BioCreative challenge evaluation workshop*, volume 1, pages 141–146.

Junnan Li, Richard Socher, and Steven C. H. Hoi. 2020. DivideMix: Learning with noisy labels as semi-supervised learning. In *8th International Confer-*

*ence on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net.

Junnan Li, Caiming Xiong, and Steven C. H. Hoi. 2021. CoMatch: Semi-supervised learning with contrastive graph regularization. In *2021 IEEE/CVF International Conference on Computer Vision, ICCV 2021, Montreal, QC, Canada, October 10-17, 2021*, pages 9455–9464. IEEE.

Xin Li and Dan Roth. 2002. Learning question classifiers. In *COLING 2002: The 19th International Conference on Computational Linguistics*.

Chen Liang, Yue Yu, Haoming Jiang, Siawpeng Er, Ruijia Wang, Tuo Zhao, and Chao Zhang. 2020. BOND: BERT-assisted open-domain named entity recognition with distant supervision. In *KDD '20: The 26th ACM SIGKDD Conference on Knowledge Discovery and Data Mining, Virtual Event, CA, USA, August 23-27, 2020*, pages 1054–1064. ACM.

Pierre Lison, Jeremy Barnes, and Aliaksandr Hubin. 2021. skweak: Weak supervision made easy for NLP. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing: System Demonstrations*, pages 337–346, Online. Association for Computational Linguistics.

Haokun Liu, Derek Tam, Muqeeth Mohammed, Jay Mohta, Tenghao Huang, Mohit Bansal, and Colin Raffel. 2022. Few-shot parameter-efficient fine-tuning is better and cheaper than in-context learning. In *Advances in Neural Information Processing Systems*.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. RoBERTa: A robustly optimized BERT pretraining approach. *CoRR*, abs/1907.11692.

Robert Logan IV, Ivana Balazevic, Eric Wallace, Fabio Petroni, Sameer Singh, and Sebastian Riedel. 2022. Cutting down on prompts and parameters: Simple few-shot learning with language models. In *Findings of the Association for Computational Linguistics: ACL 2022*, pages 2824–2835, Dublin, Ireland. Association for Computational Linguistics.

Ilya Loshchilov and Frank Hutter. 2019. Decoupled weight decay regularization. In *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*. OpenReview.net.

Yao Lu, Max Bartolo, Alastair Moore, Sebastian Riedel, and Pontus Stenetorp. 2022. Fantastically ordered prompts and where to find them: Overcoming few-shot prompt order sensitivity. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2022, Dublin, Ireland, May 22-27, 2022*, pages 8086–8098. Association for Computational Linguistics.

Andrew L. Maas, Raymond E. Daly, Peter T. Pham, Dan Huang, Andrew Y. Ng, and Christopher Potts. 2011. Learning word vectors for sentiment analysis. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies - Volume 1*, HLT '11, page 142–150, USA. Association for Computational Linguistics.

Takeru Miyato, Shin-ichi Maeda, Masanori Koyama, and Shin Ishii. 2018. Virtual adversarial training: a regularization method for supervised and semi-supervised learning. *IEEE transactions on pattern analysis and machine intelligence*, 41(8):1979–1993.

Marius Mosbach, Maksym Andriushchenko, and Dietrich Klakow. 2021. On the stability of fine-tuning BERT: Misconceptions, explanations, and strong baselines. In *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*. OpenReview.net.

Avital Oliver, Augustus Odena, Colin Raffel, Ekin Dogus Cubuk, and Ian J. Goodfellow. 2018. Realistic evaluation of deep semi-supervised learning algorithms. In *Advances in Neural Information Processing Systems 31: Annual Conference on Neural Information Processing Systems 2018, NeurIPS 2018, December 3-8, 2018, Montréal, Canada*, pages 3239–3250.

Ethan Perez, Douwe Kiela, and Kyunghyun Cho. 2021. True few-shot learning with language models. In *Advances in Neural Information Processing Systems 34: Annual Conference on Neural Information Processing Systems 2021, NeurIPS 2021, December 6-14, 2021, virtual*, pages 11054–11070.

Matthew E. Peters, Sebastian Ruder, and Noah A. Smith. 2019. To tune or not to tune? Adapting pretrained representations to diverse tasks. In *Proceedings of the 4th Workshop on Representation Learning for NLP (RepL4NLP-2019)*, pages 7–14, Florence, Italy. Association for Computational Linguistics.

Jonas Pfeiffer, Andreas Rücklé, Clifton Poth, Aishwarya Kamath, Ivan Vulić, Sebastian Ruder, Kyunghyun Cho, and Iryna Gurevych. 2020. AdapterHub: A framework for adapting transformers. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 46–54.

Sameer Pradhan, Alessandro Moschitti, Nianwen Xue, Hwee Tou Ng, Anders Björkelund, Olga Uryupina, Yuchen Zhang, and Zhi Zhong. 2013. Towards robust linguistic analysis using OntoNotes. In *Proceedings of the Seventeenth Conference on Computational Natural Language Learning*, pages 143–152, Sofia, Bulgaria. Association for Computational Linguistics.

Alexander Ratner, Stephen H. Bach, Henry Ehrenberg, Jason Fries, Sen Wu, and Christopher Ré. 2017. Snorkel: Rapid training data creation with weak supervision. *Proc. VLDB Endow.*, 11(3):269–282.

Mengye Ren, Wenyuan Zeng, Bin Yang, and Raquel Urtasun. 2018. Learning to reweight examples for robust deep learning. In *Proceedings of the 35th International Conference on Machine Learning, ICML 2018, Stockholmsmässan, Stockholm, Sweden, July 10-15, 2018*, volume 80 of *Proceedings of Machine Learning Research*, pages 4331–4340. PMLR.

Wendi Ren, Yinghao Li, Hanting Su, David Kartchner, Cassie Mitchell, and Chao Zhang. 2020. Denoising multi-source weak supervision for neural text classification. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 3739–3754, Online. Association for Computational Linguistics.

Timo Schick and Hinrich Schütze. 2022. True few-shot learning with Prompts—A real-world perspective. *Transactions of the Association for Computational Linguistics*, 10:716–731.

Jun Shu, Qi Xie, Lixuan Yi, Qian Zhao, Sanping Zhou, Zongben Xu, and Deyu Meng. 2019. Meta-weight-net: Learning an explicit mapping for sample weighting. In *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, December 8-14, 2019, Vancouver, BC, Canada*, pages 1917–1928.

Ryan Smith, Jason A. Fries, Braden Hancock, and Stephen H. Bach. 2022. Language models in the loop: Incorporating prompting into weak supervision. *CoRR*, abs/2205.02318.

Andreas Stephan, Vasiliki Kougia, and Benjamin Roth. 2022. SepLL: Separating latent class labels from weak supervision noise. In *Findings of the Association for Computational Linguistics: EMNLP 2022*, pages 3918–3929, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Michael Tänzer, Sebastian Ruder, and Marek Rei. 2022. Memorisation versus generalisation in pre-trained language models. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2022, Dublin, Ireland, May 22-27, 2022*, pages 7564–7578. Association for Computational Linguistics.

Antti Tarvainen and Harri Valpola. 2017. Mean teachers are better role models: Weight-averaged consistency targets improve semi-supervised deep learning results. In *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Workshop Track Proceedings*. OpenReview.net.

Erik F. Tjong Kim Sang and Fien De Meulder. 2003. Introduction to the CoNLL-2003 shared task: Language-independent named entity recognition. In *Proceedings of the Seventh Conference on Natural Language Learning at HLT-NAACL 2003*, pages 142–147.

Alex Warstadt, Yian Zhang, Xiaocheng Li, Haokun Liu, and Samuel R. Bowman. 2020. Learning which features matter: RoBERTa acquires a preference for linguistic generalizations (eventually). In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 217–235, Online. Association for Computational Linguistics.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander M. Rush. 2020. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 38–45, Online. Association for Computational Linguistics.

Qizhe Xie, Zihang Dai, Eduard H. Hovy, Thang Luong, and Quoc Le. 2020. Unsupervised data augmentation for consistency training. In *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*.

Yue Yu, Simiao Zuo, Haoming Jiang, Wendi Ren, Tuo Zhao, and Chao Zhang. 2021. Fine-tuning pre-trained language model with weak supervision: A contrastive-regularized self-training approach. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2021, Online, June 6-11, 2021*, pages 1063–1077. Association for Computational Linguistics.

Elad Ben Zaken, Yoav Goldberg, and Shauli Ravfogel. 2022. BitFit: Simple parameter-efficient fine-tuning for transformer-based masked language-models. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers), ACL 2022, Dublin, Ireland, May 22-27, 2022*, pages 1–9. Association for Computational Linguistics.

Bowen Zhang, Yidong Wang, Wenxin Hou, Hao Wu, Jindong Wang, Manabu Okumura, and Takahiro Shinozaki. 2021a. FlexMatch: Boosting semi-supervised learning with curriculum pseudo labeling. In *Advances in Neural Information Processing Systems 34: Annual Conference on Neural Information Processing Systems 2021, NeurIPS 2021, December 6-14, 2021, virtual*, pages 18408–18419.

Chiyuan Zhang, Samy Bengio, Moritz Hardt, Benjamin Recht, and Oriol Vinyals. 2017. Understanding deep learning requires rethinking generalization. In *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*. OpenReview.net.

Jieyu Zhang, Cheng-Yu Hsieh, Yue Yu, Chao Zhang, and Alexander Ratner. 2022. A survey on programmatic weak supervision. *CoRR*, abs/2202.05433.

Jieyu Zhang, Yue Yu, NameError, Yujing Wang, Yaming Yang, Mao Yang, and Alexander Ratner. 2021b. WRENCH: A comprehensive benchmark for weak supervision. In *Proceedings of the Neural Information Processing Systems Track on Datasets and Benchmarks 1, NeurIPS Datasets and Benchmarks 2021, December 2021, virtual*.

Xiang Zhang, Junbo Zhao, and Yann LeCun. 2015. Character-level convolutional networks for text classification. *Advances in neural information processing systems*, 28.

Zihao Zhao, Eric Wallace, Shi Feng, Dan Klein, and Sameer Singh. 2021. Calibrate before use: Improving few-shot performance of language models. In *Proceedings of the 38th International Conference on Machine Learning, ICML 2021, 18-24 July 2021, Virtual Event*, volume 139 of *Proceedings of Machine Learning Research*, pages 12697–12706. PMLR.

Guoqing Zheng, Ahmed Hassan Awadallah, and Susan T. Dumais. 2021. Meta label correction for noisy label learning. In *Thirty-Fifth AAAI Conference on Artificial Intelligence, AAAI 2021, Thirty-Third Conference on Innovative Applications of Artificial Intelligence, IAAI 2021, The Eleventh Symposium on Educational Advances in Artificial Intelligence, EAAI 2021, Virtual Event, February 2-9, 2021*, pages 11053–11061. AAAI Press.

Guoqing Zheng, Giannis Karamanolakis, Kai Shu, and Ahmed Awadallah. 2022a. WALNUT: A benchmark on semi-weakly supervised learning for natural language understanding. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 873–899, Seattle, United States. Association for Computational Linguistics.

Yanan Zheng, Jing Zhou, Yujie Qian, Ming Ding, Chonghua Liao, Li Jian, Ruslan Salakhutdinov, Jie Tang, Sebastian Ruder, and Zhilin Yang. 2022b. FewNLU: Benchmarking state-of-the-art methods for few-shot natural language understanding. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2022, Dublin, Ireland, May 22-27, 2022*, pages 501–516. Association for Computational Linguistics.

Jiawei Zhou, Xiaoguang Li, Lifeng Shang, Lan Luo, Ke Zhan, Enrui Hu, Xinyu Zhang, Hao Jiang, Zhao Cao, Fan Yu, Xin Jiang, Qun Liu, and Lei Chen. 2022. Hyperlink-induced pre-training for passage retrieval in open-domain question answering. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2022, Dublin, Ireland, May 22-27, 2022*, pages 7135–7146. Association for Computational Linguistics.

Dawei Zhu, Michael A. Hedderich, Fangzhou Zhai, David Ifeoluwa Adelani, and Dietrich Klakow. 2022. Is BERT robust to label noise? A study on learning with noisy labels in text classification. In *Proceedings of the Third Workshop on Insights from Negative Results in NLP, Insights@ACL 2022, Dublin, Ireland, May 26, 2022*, pages 62–67. Association for Computational Linguistics.

Dawei Zhu, Xiaoyu Shen, Michael A. Hedderich, and Dietrich Klakow. 2023. Meta self-refinement for robust learning with weak supervision. In *Proceedings of the 17th Conference of the European Chapter of the Association for Computational Linguistics, EACL 2023, Dubrovnik, Croatia, May 2-6, 2023*, pages 1043–1058. Association for Computational Linguistics.

## A Datasets

In the following, we give a more comprehensive description of the datasets used. A subset of the commonly used WRENCH (Zhang et al., 2021b) benchmark is used, covering various aspects such as task type, coverage and dataset size. There is a total of four classification, two relation extraction and two sequence labeling datasets. See Table 2 for a detailed set of data statistics.

**AGNews** (Zhang et al., 2015) is a topic classification dataset. The task is to classify news articles into four topics, namely world, sports, business and Sci-Fi/technology. Each labeling function is composed of multiple keywords to search for. The number of keywords differs from a few up to dozens.

**IMDb** (Maas et al., 2011) is a dataset of movie reviews sampled from the IMDb website. The task is binary sentiment analysis. The labeling functions are composed of keyword searches and regular expressions.

**Yelp** (Zhang et al., 2015) is another sentiment analysis dataset, containing crowd-sourced business reviews. The labeling functions are created using keywords and a lexicon-based sentiment analysis library.

**TREC** (Li and Roth, 2002) is a question classification dataset, i.e., it asks what type of response is expected. The labels are abbreviation, description and abstract concepts, entities, human beings, locations or numeric values. The labeling functions are created using regular expressions and make a lot of use of question words such as "what", "where" or "who".

**SemEval** (Hendrickx et al., 2010) is a relation classification dataset, using nine relation types. Examples for relation labels are cause-effect, entity-origin or message-topic. Labeling functions are created using entities within a regular expression.

**ChemProt** (Krallinger et al., 2017) is another relation classification dataset, focusing on chemical research literature. It contains ten different types of relations, for example chemical-protein relations such as “biological properties upregulator”. The labeling functions are created using rules.

**CoNLL-03** (Tjong Kim Sang and De Meulder, 2003) is a named entity recognition (NER) dataset, with labels for the entities "person", "location", "organization", and "miscellaneous". Labeling functions are built using previously trained keywords, regular expressions and NER models.

**OntoNotes 5.0** (Pradhan et al., 2013) is a another NER dataset, using more fine-grained entities as CoNLL-03. Here, a subset of the CoNLL weak labeling sources is combined with keyword and regular expression based weak labeling sources.

## B Labeling functions

Weak labeling sources are often abstracted as labeling functions and vary in aspects such as coverage, precision, or overlap (Ratner et al., 2017; Karamanolakis et al., 2021). To showcase how the weak labeling process works, a selection of examples of labeling functions is presented. More specifically, we provide examples of rules for the two classification datasets IMDb (Table 3) and TREC (Table 4), the relation classification dataset SemEval (Table 5) and the NER dataset CoNLL-03 (Table 6).

## C Overall implementation details

This section summarizes the overall implementation details of WSL approaches used in our paper. Refer to Appendix D for hyperparameter configurations of PEFT approaches. We use the PyTorch framework[11] to implement all approaches discussed in the paper. Hugging Face (Wolf et al., 2020) is used for downloading and training the RoBERTa-base model. AdapterHub (Pfeiffer et al., 2020) is used for implementing parameter-efficient fine-tuning.

**Hyperparameters** In this paper, we implemented five WSL methods: FT (Devlin et al., 2019), L2R (Ren et al., 2018), MLC (Zheng et al., 2021), BOND (Liang et al., 2020), and COSINE (Yu et al., 2021). We report the search ranges of the hyperparameters in Table 7.

We do not search for batch size as we find it has minor effects on the final performance. Instead, a batch size of 32 is used across experiments. Also, RoBERTa-base (Liu et al., 2019) is used as the backbone PLM and AdamW (Loshchilov and Hutter, 2019) is the optimizer used across all methods.

**Computing infrastructure and training cost** We use Nvidia V100-32 GPUs for training deep learning models. All WSL approaches studied in

[11] https://pytorch.org/

| Dataset | Task | #Classes | #LFs | %Ovr. Coverage | Avg. over labeling functions (LFs) | | | | MV | #Train | #Dev | #Test |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | | | | %Coverage | %Overlap | %Conflict | %Prec. | | | | |
| AGNews | News Class. | 4 | 9 | 69.08 | 10.34 | 5.05 | 2.43 | 81.66 | 81.23 | 96,000 | 12,000 | 12,000 |
| IMDb | Movie Sentiment Class. | 2 | 5 | 87.58 | 23.60 | 11.60 | 4.50 | 69.88 | 73.86 | 20,000 | 2,500 | 2,500 |
| Yelp | Business Sentiment Class. | 2 | 8 | 82.78 | 18.34 | 13.58 | 4.94 | 73.05 | 73.31 | 30,400 | 3,800 | 3,800 |
| TREC | Question Class. | 6 | 68 | 95.13 | 2.55 | 1.82 | 0.84 | 75.92 | 62.58 | 4,965 | 500 | 500 |
| SemEval | Web Text Relation Class. | 9 | 164 | 100.00 | 0.77 | 0.32 | 0.14 | 97.69 | 77.33 | 1,749 | 200 | 692 |
| ChemProt | Chemical Relation Class. | 10 | 26 | 85.62 | 5.93 | 4.40 | 3.95 | 46.65 | 55.12 | 12,861 | 1,607 | 1,607 |
| CoNLL-03 | English News NER | 4 | 16 | 100 | 100 | 4.30 | 1.44 | 72.19 | 60.38 | 14,041 | 3250 | 3453 |
| OntoNotes 5.0 | Multi-Domain NER | 18 | 17 | 100 | 100 | 1.55 | 0.54 | 54.84 | 58.92 | 115,812 | 5,000 | 22,897 |

Table 2: Detailed data statistics. Note that 'Class.' is an abbreviation for classification. Coverage is the amount of samples a labeling function (LF) matches. For NER datasets, labeling functions return an entity or "O" thus coverage is always 100%. Overlap asks how many samples have at least 2 matching labeling functions. MV (majority vote) performance is given as F1-score for the NER datasets and as accuracy on the test set otherwise.

| Label | Labeling Function |
|---|---|
| POS | beautiful, handsome, talented |
| NEG | than this, than the film, than the movie |
| POS | .*(highly\|do\|would\|definitely\|certainly\|strongly\|i\|we).*(recommend\|nominate).* |
| POS | .*(high\|timeless\|priceless\|HAS\|great\|real\|instructive).*(value\|quality\|meaning\|significance).* |

Table 3: Examples of two keyword based and two regular expression based rules for the IMDb dataset.

this paper can fit into one single GPU. We report the training time of the WSL methods in Table 8.

# D Training with clean samples

## D.1 Methods and implementation details

In Section 6, we apply four (parameter-efficient) fine-tuning approaches to train models on clean validation sets. Since we do not have extra data for model selection, we choose a fixed set of hyperparameters for all datasets. In the following we briefly introduce the fine-tuning approaches, together with their hyperparameter configurations.

- Vanilla fine-tuning (Devlin et al., 2019; Liu et al., 2019) is the standard fine-tuning approaches for pre-trained language models. It works by adding a randomly initialized classifier on top of the pre-trained model and training it together with all other model parameters. We use a fixed learning rate of $2e^{-5}$ in all experiments.

- Adapter-based fine-tuning (Houlsby et al., 2019) adds additional feed-forward layers called adapters to each layer of the pre-trained language model. During fine-tuning, we only update the weights of these adapter layers and keep all other parameters *frozen* at their pre-trained values. We use a fixed learning rate of $2e^{-5}$ in all experiments. The reduction factor is set to 16.

- BitFit (Zaken et al., 2022) updates only the bias parameters of every layer and keeps all other weights frozen. Despite its simplicity it has been demonstrated to achieve similar results to adapter-based fine-tuning. We use a fixed learning rate of $1e^{-4}$ in all experiments.

- LoRA (Hu et al., 2022) is a recently proposed adapter-based fine-tuning method which uses a low-rank bottleneck architecture in each of the newly added feed-forward networks. The motivation here is to perform a low rank update to the model during fine-tuning. We use a fixed learning rate of $2e^{-5}$ in all experiments. The $\alpha$ value used in LoRa is fixed to 16.

In all experiments, the batch size used in all fine-tuning approaches is 32. The optimizer is AdamW (Loshchilov and Hutter, 2019).

## D.2 Training on the full validation sets

In addition to training sets, the WRENCH (Zhang et al., 2021b) benchmark provides a validation set for each of its tasks. The validation sets are cleanly annotated and typically range in size from 5% to 25% of the weakly annotated training sets. Although such validation size is reasonable for fully supervised learning, we suspect that it is exorbitant in the sense that it provides a significantly better training signal for models than the weakly annotated training set. Thus we compare the performance of recent WSL approaches that access

| Label | Labeling Function |
|---|---|
| ABBREVIATION | ( \|^)(what\|what)[^\w]* (\w+ ){0,1}(does\|does)[^\w]* ([^\s]+ )*(stand for)[^\w]*( \|$) |
| DESCRIPTION | ( \|^)(explain\|describe\|how\|how)[^\w]* (\w+ ){0,1}(can\|can)[^\w]*( \|$) |
| ENTITY | ( \|^)(which\|what\|what)[^\w]* ([^\s]+ )*(organization\|trust\|company\|company)[^\w]*( \|$) |
| HUMAN | ( \|^)(who\|who)[^\w]*( \|$) |
| LOCATION | ( \|^)(which\|what\|where\|where)[^\w]* ([^\s]+ )*(situated\|located\|located)[^\w]*( \|$) |
| NUMERIC | ( \|^)(by how\|how\|how)[^\w]* (\w+ ){0,1}(much\|many\|many)[^\w]*( \|$) |

Table 4: Rules for the TREC dataset. For each label a representative labeling function is given.

| Label | Labeling Function |
|---|---|
| Cause-Effect(e1,e2) | SUBJ-O caused OBJ-O |
| Component-Whole(e1,e2) | SUBJ-O is a part of the OBJ-O |
| Content-Container(e1,e2) | SUBJ-O was contained in a large OBJ-O |
| Entity-Destination(e1,e2) | SUBJ-O into OBJ-O |
| Entity-Origin(e1,e2) | SUBJ-O emerged from the OBJ-O |
| Instrument-Agency(e2,e1) | SUBJ-O took the OBJ-O |
| Member-Collection(e2,e1) | SUBJ-O of different OBJ-O |
| Message-Topic(e1,e2) | SUBJ-O states that the OBJ-O |
| Product-Producer(e1,e2) | SUBJ-O created by the OBJ-TITLE |

Table 5: One labeling function for each label of the SemEval dataset. Here e1 and e2 are entities which are already available in the dataset.

| Label | Labeling Function |
|---|---|
| PERSON | RegEx searching list one of 7559 first names, followed by an upper-cased word |
| LOCATION | List of 15205 places |
| ORGANIZATION | WTO, Starbucks, mcdonald, google, Baidu, IBM, Sony, Nikon |
| MISCELLANEOUS | List of countries, languages, events and facilities |

Table 6: For each label, one labeling function of the CoNLL-03 dataset is displayed.

| Hyperparameter | Search Range |
|---|---|
| Learning rate | 2e-5, 3e-5, 5e-5 |
| Warm-up steps | 50, 100, 200 |

(a) FT (for both training on clean or weak labels)

| Hyperparameter | Search Range |
|---|---|
| Learning rate | 2e-5, 3e-5, 5e-5 |
| Meta-learning rate | 1e-4, 2e-5, 1e-5 |

(b) L2R

| Hyperparameter | Search Range |
|---|---|
| Learning rate | 2e-5, 3e-5, 5e-5 |
| Meta-learning rate | 1e-4, 2e-5, 1e-5 |
| hdim | 512, 768 |

(c) MLC

| Hyperparameter | Search Range |
|---|---|
| Learning rate | 2e-5, 3e-5, 5e-5 |
| $T_1$ | 5000 |
| $T_2$ | 5000 |
| $T_3$ | 50, 100, 300, 500 |
| Confidence threshold | 0.1, 0.3, 0.5, 0.7, 0.8, 0.9 |

(d) BOND

| Hyperparameter | Search Range |
|---|---|
| Learning rate | 2e-5, 3e-5, 5e-5 |
| $T_1$ | 5000 |
| $T_2$ | 5000 |
| $T_3$ | 50, 100, 300, 500 |
| Distance measure | cosine |
| Regularization factor | 0.05 0.1 0.2 |
| Confidence threshold | 0.1, 0.3, 0.5, 0.7, 0.8, 0.9 |

(e) COSINE

Table 7: The search range of the hyperparameters of the five WSL approaches considered in the paper. For BOND and COSINE, we set $T_1$ and $T_2$ to constant values, because we stop training once early-stopping is triggered.

| | AGNews | IMDb | Yelp | TREC | SemEval | ChemProt | CoNLL-03 | OntoNotes 5.0 |
|---|---|---|---|---|---|---|---|---|
| FT | 0.2 | 0.2 | 0.2 | 0.1 | 0.1 | 0.2 | 0.2 | 0.5 |
| L2R | 2.0 | 1.2 | 1.5 | 0.3 | 0.3 | 0.4 | 0.9 | 1.2 |
| MLC | 1.2 | 0.8 | 1.2 | 0.3 | 0.2 | 0.5 | 1.2 | 1.0 |
| BOND | 0.5 | 0.2 | 0.5 | 0.1 | 0.1 | 0.2 | 0.4 | 1.1 |
| COSINE | 0.6 | 0.2 | 0.6 | 0.2 | 0.2 | 0.3 | 0.5 | 1.5 |

Table 8: Running time in hours of each WSL method when trained on a weakly labeled training set. Since we also track the validation and test performance during training, the training time reported here actually overestimates the training time required for each method.

both the training and validation sets with a model that is directly fine-tuned on the validation set. The following WSL methods are included in this experiment: L2R (Ren et al., 2018), MetaWN (Shu et al., 2019), BOND (Liang et al., 2020), Denoise (Ren et al., 2020), MLC (Zheng et al., 2021), and COSINE (Yu et al., 2021). Following prior work, we select the best set of hyperparameters via the validation set when applying the WSL methods. Also, early-stopping based on the validation performance is applied. In contrast, the direct fine-tuning baseline uses a fixed set of hyperparameters across all datasets, and no early-stopping is applied (same configuration as in Appendix D.1). We train this baseline for 6000 steps. In all cases, the training losses converged much earlier than 6000 steps, but we deliberately kept training for longer to show that the good performance achieved by this baseline is not due to any fine-grained configurations. As shown in Figure 1, this simple baseline outperforms all the WSL methods in all but one case.

### D.3 Extended comparison of training on clean data and validation for WSL approaches

In Section 6, standard fine-tuning (FT) and multiple parameter-efficient fine-tuning (PEFT) are compared with the competitive WSL method COSINE. In this section, we provide additional plots which show the same comparison with the other WSL methods examined in this work, namely L2R, MLC, and BOND. We report average performance (Acc. and F1 in %) difference between (parameter-efficient) fine-tuning methods and the specific WSL method for varying number of clean samples. The overall tendency is consistent with the results in Section 6: WSL methods perform well on a small amount of clean labeled data but PEFT outperforms WSL methods with an increasing amount of clean labeled data.

## E Additional baselines that combine weak and clean data during training

Besides CFT we also explored two simple baselines that combine both the cleanly and weakly annotated data in training:

1. **WC$_{\text{mix}}$**: it mixes the clean data into the weakly labeled training set. We then fine-tune a PLM on this combined dataset.

2. **WC$_{\text{batch}}$**: in each batch, we mix the weakly and cleanly labeled data at a ratio of 50:50. This makes sure that the model can access clean samples in each batch.

We compared these two baselines with CFT, the results are shown in Figure 9. It can be seen that when the same amount of data is accessed, CFT outperforms the two baselines in most cases, sometimes by a large margin.

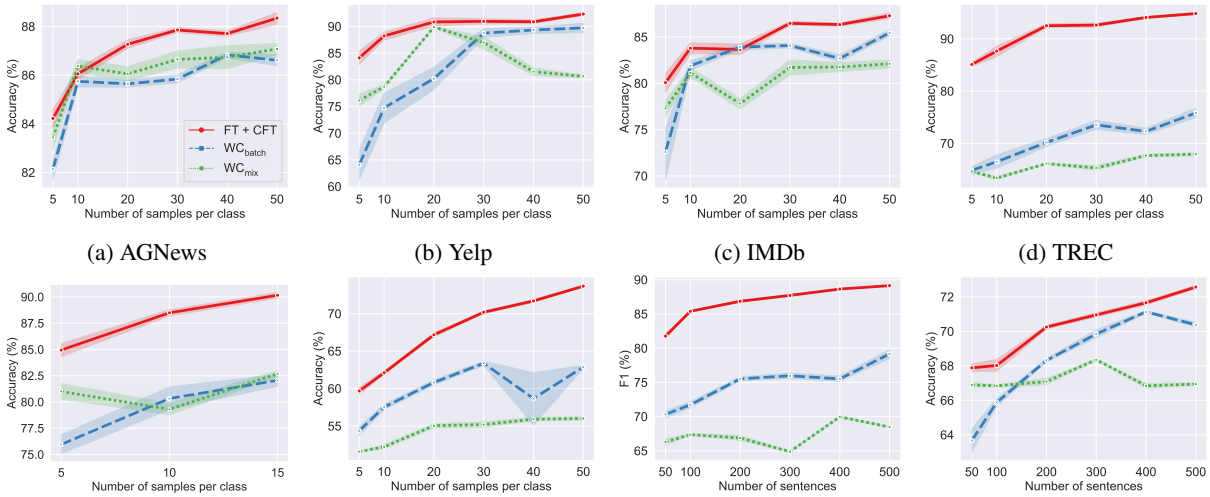
## F Additional plots on CFT with different numbers of clean samples

We show further plots of experiments in Section 7 with different numbers of clean samples in Figure 10. More specifically, it shows the results for selecting $N \in \{10, 20, 30, 40\}$ clean samples per class from the clean validation set for classification and $N \in \{100, 200, 300, 400\}$ for NER tasks. These results corroborate the analysis presented in Section 7.

## G CFT with different PLMs and agreement ratios

We provide additional plots of the experiments mentioned in Section 8 on more datasets. Figure 11 shows the performance of CFT using different PLMs during training and Figure 12 shows the performance when the number of clean samples and the agreement ratio is varied.

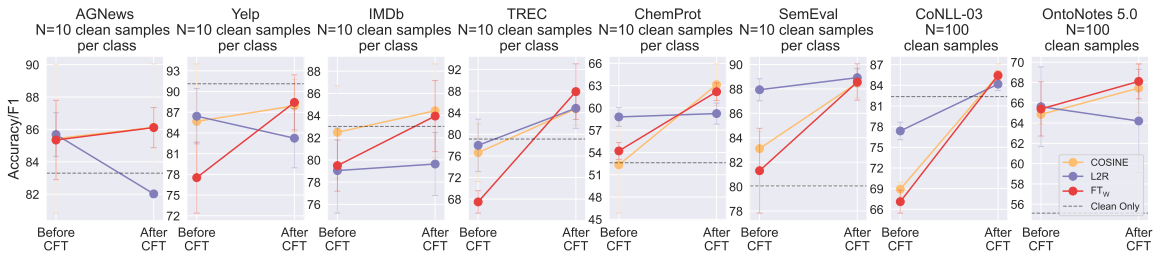

Figure 8: Performance difference of (parameter-efficient) fine-tuning approaches (FT, LoRA, BitFit and Adapter) with WSL apporaches (L2R, MLC, BOND and COSINE), using varying amounts of clean data. We use the subscript "C" (e.g., $\text{FT}_\text{C}$) to indicate that the fine-tuning approaches are applied on clean data.



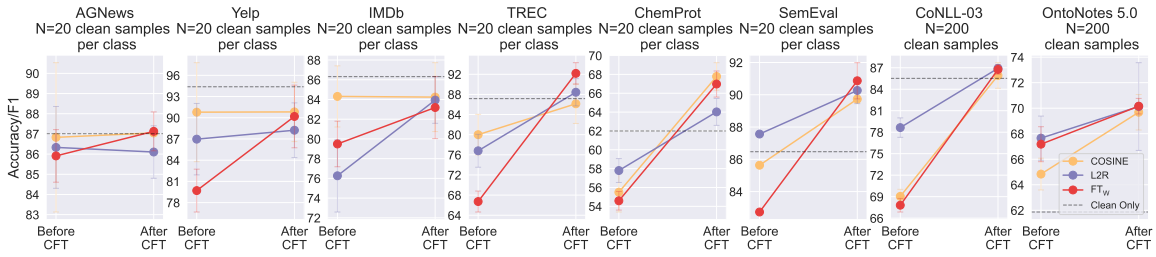

(a) AGNews (b) Yelp (c) IMDb (d) TREC

(e) SemEval (f) ChemProt (g) CoNLL-03 (h) OntoNotes 5.0

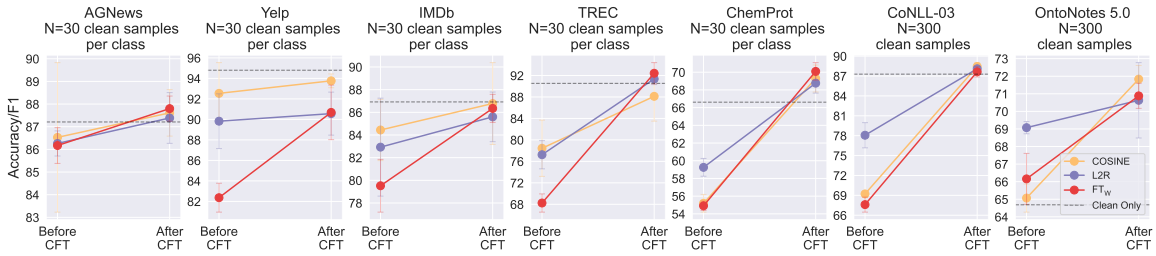
Figure 9: Performance vs. number of clean samples. In most cases, CFT outperforms the other two baselines, $\text{WC}_{\text{batch}}$ and $\text{WC}_{\text{mix}}$, by a considerable margin.



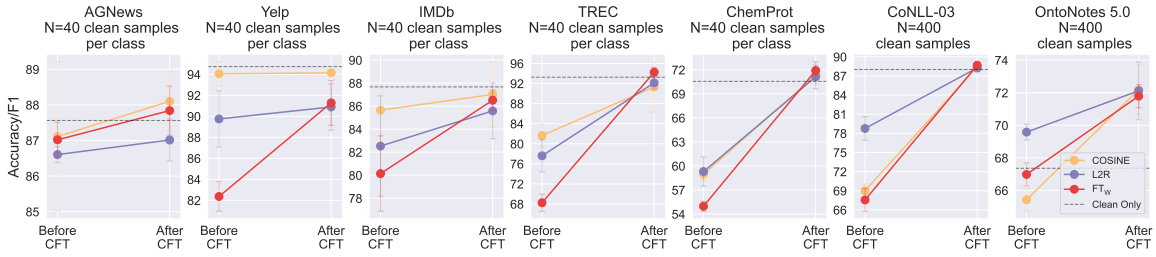

(a) $N = 10$ samples per class ($N = 100$ sentences on NER)




(b) $N = 20$ samples per class ($N = 200$ sentences on NER)

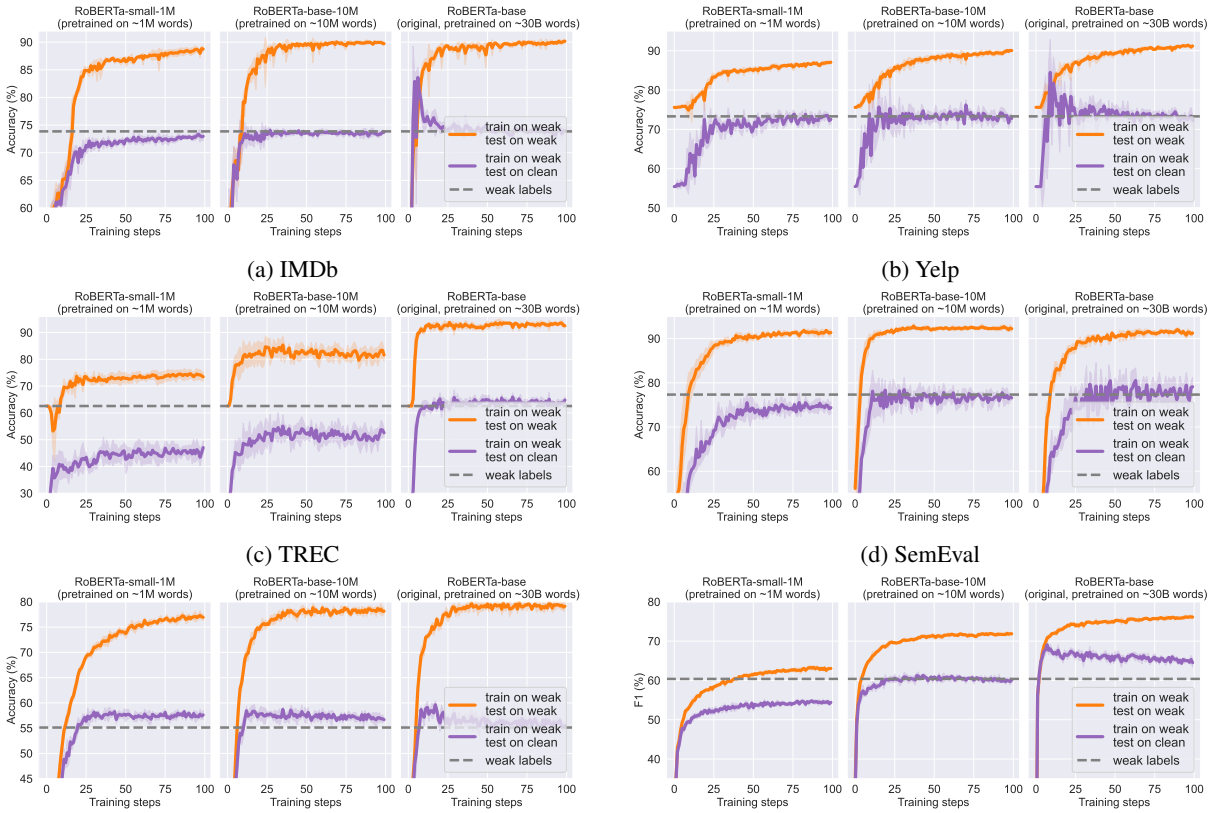



(c) $N = 30$ samples per class ($N = 300$ sentences on NER)




(d) $N = 40$ samples per class ($N = 400$ sentences on NER)

Figure 10: Performance difference before and after applying CFT to WSL methods. For text classification and relation extraction tasks, we subsample $N \in \{5, 10, 20, 30, 40, 50\}$ examples from the validation set. For NER, we subsample $N \in \{50, 100, 200, 300, 400, 500\}$. On SemEval, the original validation set is small, and sampling more than 20 samples per class is not possible. The figure shows that the performance gap between the simple baseline $\text{FT}_\text{W}$ and COSINE/L2R becomes much smaller after CFT, suggesting that we may not require sophisticated WSL methods to achieve good generalization.

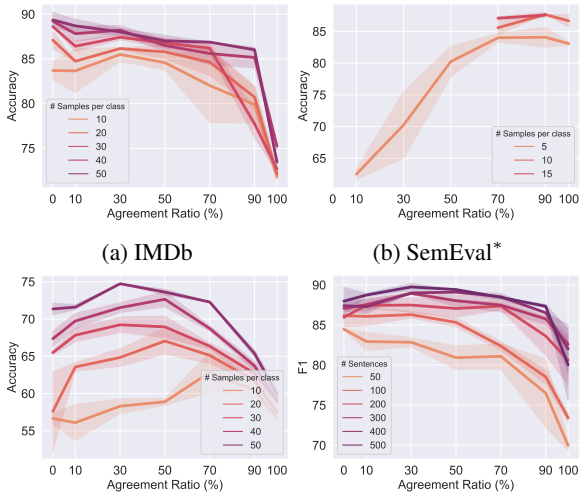



(a) IMDb (b) Yelp

(c) TREC (d) SemEval

(e) ChemProt (f) CoNLL-03

Figure 11: **Performance curves of different PLMs during training**. PLMs are trained on weak labels and evaluated on both clean and weakly labeled test sets. Pre-training on larger corpora improves performance on the clean distribution.




(a) IMDb (b) SemEval*

(c) ChemProt (d) CoNLL-03

Figure 12: **Model performance varying the number of clean samples $N$ and agreement ratio $\alpha$.** Large values of $\alpha$ generally cause a substantial performance drop. *: Certain combinations of $\alpha$ and $N$ are not feasible because the validation set lacks samples with clean and weak labels that coincide or differ.

## ACL 2023 Responsible NLP Checklist

### A For every submission:

☑ A1. Did you describe the limitations of your work?
*Section after conclusion, no section number*

☐ A2. Did you discuss any potential risks of your work?
*Not applicable. Left blank.*

☑ A3. Do the abstract and introduction summarize the paper's main claims?
*Left blank.*

☒ A4. Have you used AI writing assistants when working on this paper?
*Left blank.*

### B ☒ Did you use or create scientific artifacts?

*Left blank.*

☐ B1. Did you cite the creators of artifacts you used?
*Not applicable. Left blank.*

☐ B2. Did you discuss the license or terms for use and / or distribution of any artifacts?
*Not applicable. Left blank.*

☐ B3. Did you discuss if your use of existing artifact(s) was consistent with their intended use, provided that it was specified? For the artifacts you create, do you specify intended use and whether that is compatible with the original access conditions (in particular, derivatives of data accessed for research purposes should not be used outside of research contexts)?
*Not applicable. Left blank.*

☐ B4. Did you discuss the steps taken to check whether the data that was collected / used contains any information that names or uniquely identifies individual people or offensive content, and the steps taken to protect / anonymize it?
*Not applicable. Left blank.*

☐ B5. Did you provide documentation of the artifacts, e.g., coverage of domains, languages, and linguistic phenomena, demographic groups represented, etc.?
*Not applicable. Left blank.*

☐ B6. Did you report relevant statistics like the number of examples, details of train / test / dev splits, etc. for the data that you used / created? Even for commonly-used benchmark datasets, include the number of examples in train / validation / test splits, as these provide necessary context for a reader to understand experimental results. For example, small differences in accuracy on large test sets may be significant, while on small test sets they may not be.
*Not applicable. Left blank.*

### C ☑ Did you run computational experiments?

*Sec 4-8*

☑ C1. Did you report the number of parameters in the models used, the total computational budget (e.g., GPU hours), and computing infrastructure used?
*Sec 4-8 + appendix*

*The Responsible NLP Checklist used at ACL 2023 is adopted from NAACL 2022, with the addition of a question on AI writing assistance.*

☑ C2. Did you discuss the experimental setup, including hyperparameter search and best-found hyperparameter values?
*Sec 4-8 + appendix*

☑ C3. Did you report descriptive statistics about your results (e.g., error bars around results, summary statistics from sets of experiments), and is it transparent whether you are reporting the max, mean, etc. or just a single run?
*Sec 4-8 + appendix*

☑ C4. If you used existing packages (e.g., for preprocessing, for normalization, or for evaluation), did you report the implementation, model, and parameter settings used (e.g., NLTK, Spacy, ROUGE, etc.)?
*Sec 4-8 + appendix*

**D ☒ Did you use human annotators (e.g., crowdworkers) or research with human participants?**

*Left blank.*

☐ D1. Did you report the full text of instructions given to participants, including e.g., screenshots, disclaimers of any risks to participants or annotators, etc.?
*Not applicable. Left blank.*

☐ D2. Did you report information about how you recruited (e.g., crowdsourcing platform, students) and paid participants, and discuss if such payment is adequate given the participants' demographic (e.g., country of residence)?
*Not applicable. Left blank.*

☐ D3. Did you discuss whether and how consent was obtained from people whose data you're using/curating? For example, if you collected data via crowdsourcing, did your instructions to crowdworkers explain how the data would be used?
*Not applicable. Left blank.*

☐ D4. Was the data collection protocol approved (or determined exempt) by an ethics review board?
*Not applicable. Left blank.*

☐ D5. Did you report the basic demographic and geographic characteristics of the annotator population that is the source of the data?
*Not applicable. Left blank.*